SKP8CMINI
Tutorial 1

Introduction

Renesas Technology America Inc.

Overview

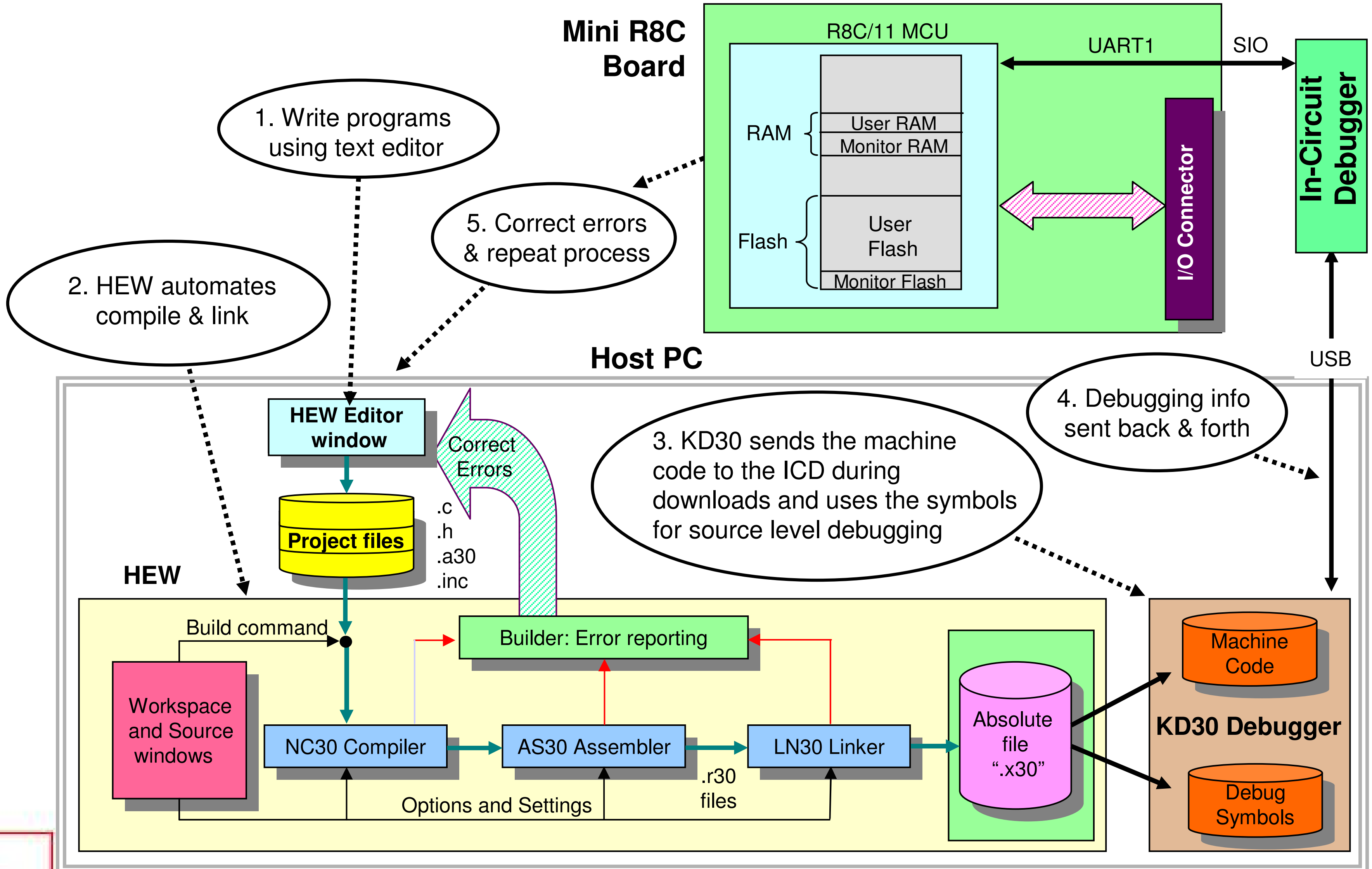
The following tutorial provides an introduction to the Mini R8C SKP. It explains the basic development environment; how to develop and debug programs using HEW (High Performance Embedded Workshop) and KD30, and how to work with existing example projects.

To get the most out of the Starter Kit, check out the references at the end of this tutorial.

Note: ***This tutorial assumes the user has done the following:***

1. ***Followed the 'Quick Start Guide'***
2. ***Installed the SKP files, examples, and software tools in the default directories.***

The Development Process



SKP8CMINI Connectivity

To PC

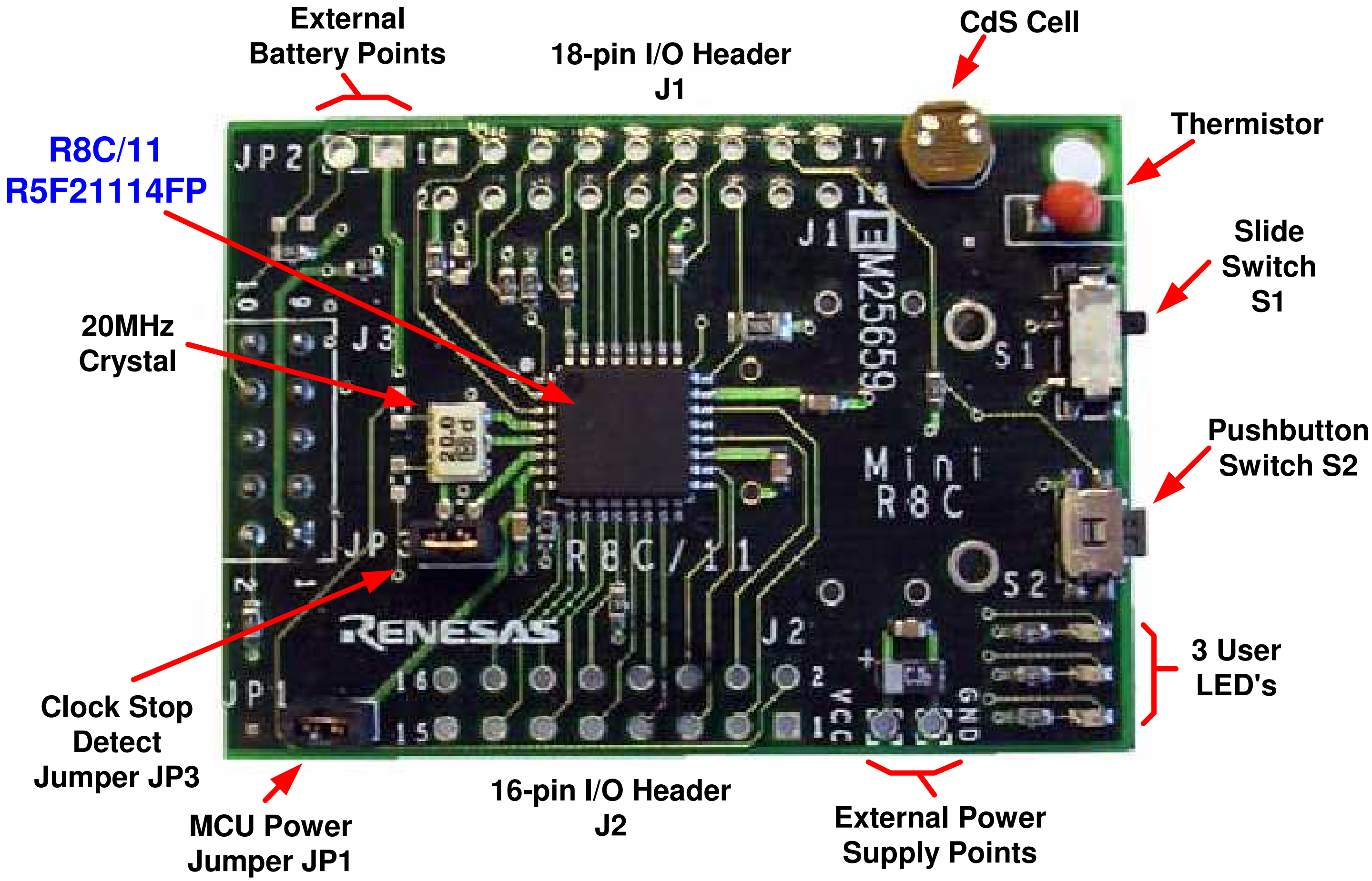


USB cable

ICD (In-Circuit Debugger)

Mini R8C board

Mini R8C Board



Mini R8C Board Features

R8C/11 (R5F21114FP) MCU

- 20MHz Operating Frequency at 3.0V – 5.5V, 10MHz Operating Frequency at 2.7V – 5V
- 16kB Flash ROM, 1kB RAM
- 24 GPIO including –
- 4 Key-on Wakeup Inputs
- 3 8-bit and 1 16-bit Timers plus a Watchdog Timer
- 12-channel 10-bit ADC
- 2 SIO – 1 Clock Sync + UART, 1 UART
- Voltage Detect and Oscillation Stop Detection
- Clock sources: Main (Xin), Ring oscillator (Low and High speed)

Onboard Features

- 3 LEDs (Red, Yellow, Green)
- 2 Switches – 1 slider, 1 pushbutton
- 2 Sensors - Thermistor and CdS cell on two A/D inputs
- Jumpers for Icc measurements and Clock Stop Detect
- I/O available on expansion port headers

ICD (RTA-FoUSB-MON)

The ICD (In-Circuit Debugger) provides power and a USB interface to the Host PC and communicates commands and data to and from the Mini R8C board via a synchronous serial interface.

As a debugging tool (during program debug), the ICD + KD30 downloads a small kernel (or ROM Monitor) program with the user program to the Mini R8C board . This kernel provides a communication interface between the R8C/11 MCU and the ICD + KD30 Debugger application on MCU status. While the kernel uses some resources of the R8C/11, the operation of the ICD is transparent to the user's program.

As a programming tool, the ICD + Flash-over-USB™(FoUSB) Programmer can be used to download user programs to the R8C/11 MCU on the Mini R8C board and many other Renesas' flash MCU's (the ICD will support other Renesas flash MCU's by downloading an MCU Monitor Image (MMI) file for a particular MCU thru KD30 or FoUSB Programmer).

NOTE: The kernel is only downloaded with the user program when using KD30 Debugger but NOT the FoUSB Programmer.

Development Tools

HEW

An Integrated Development Environment (IDE) that invokes all necessary software for building your project

KD30

PC software that communicates with the ROM Monitor Program (in flash on the MCU) for program debug

NC30 R8C/Tiny Version

C-compiler (limited version of NC30). Conforms to ANSI C standards (see release notes on limitations)

AS30

Relocatable Assembler

Supports structured language and a wide variety of macro instructions

Flash-over-USB™ Programmer

Flash programmer for Renesas Flash MCU’s.

HEW Overview

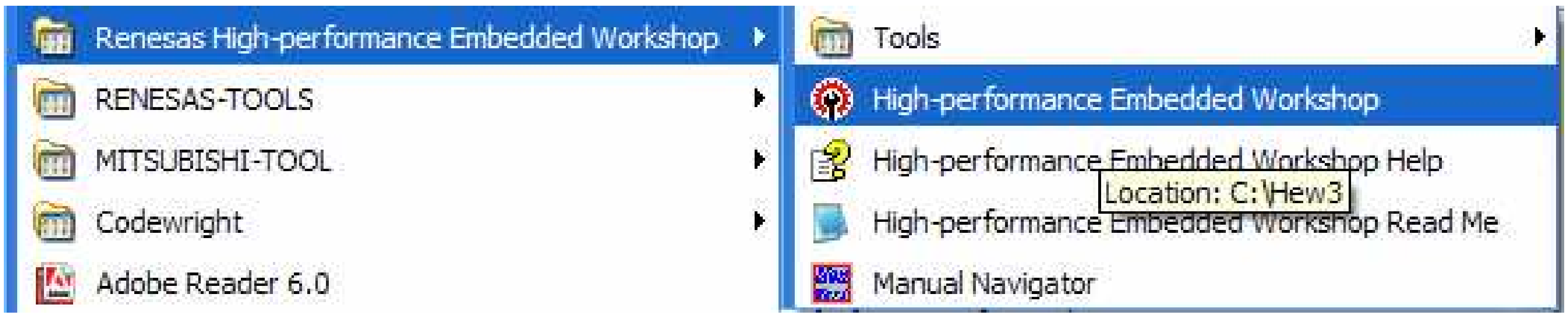
HEW is an acronym for High-performance Embedded Workshop.

When writing a microcontroller (or any computer) program, the program is usually split into multiple files to make it easier to read and understand.

While exactly how the files are organized is up to the programmer, typically, the code is split up in a logical manner into various files (e.g. math functions in one file, serial port drivers in another, etc).

After all the files in a **project** are compiled and assembled, a **linker** combines all the files into a single file. These steps can be tedious and repetitive. To make the process simple, we use an **Integrated Development Environment (IDE**) called **HEW**.

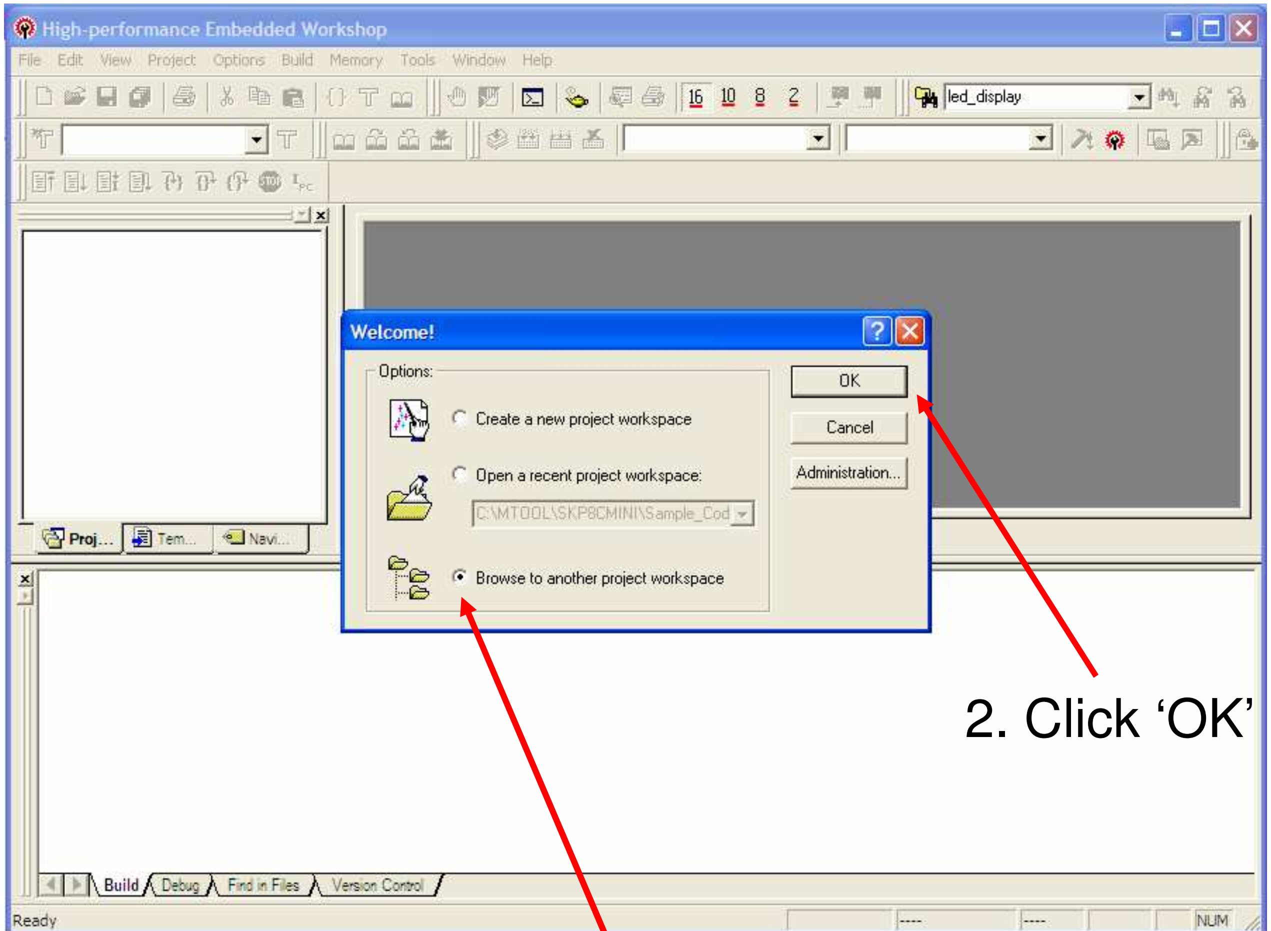
Start HEW



From the Windows Start menu, click on

Programs > Renesas High-performance Embedded Workshop>
High-performance Embedded Workshop

Open a HEW Workspace (1/3)



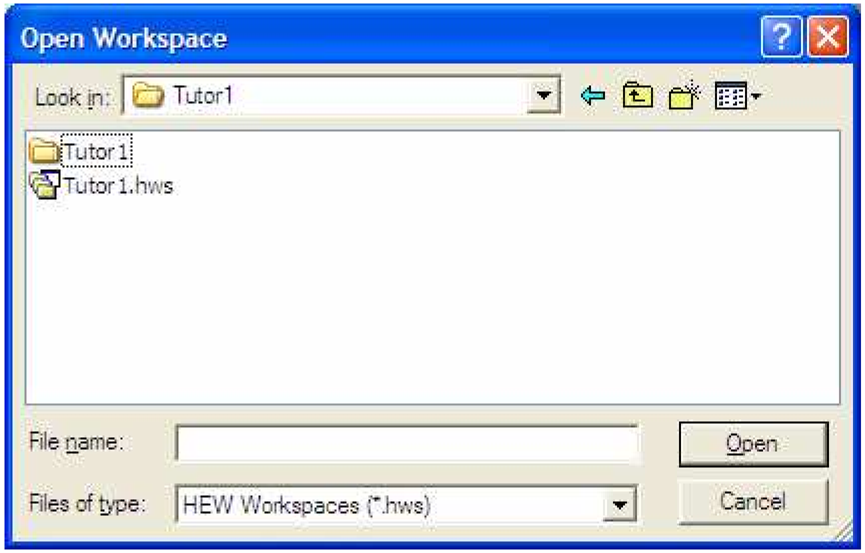
2. Click 'OK' button

1. After HEW opens, from the Welcome dialog box, select 'Browse to another project workspace' option, then click OK.

RENESAS

Open a HEW Workspace (2/3)

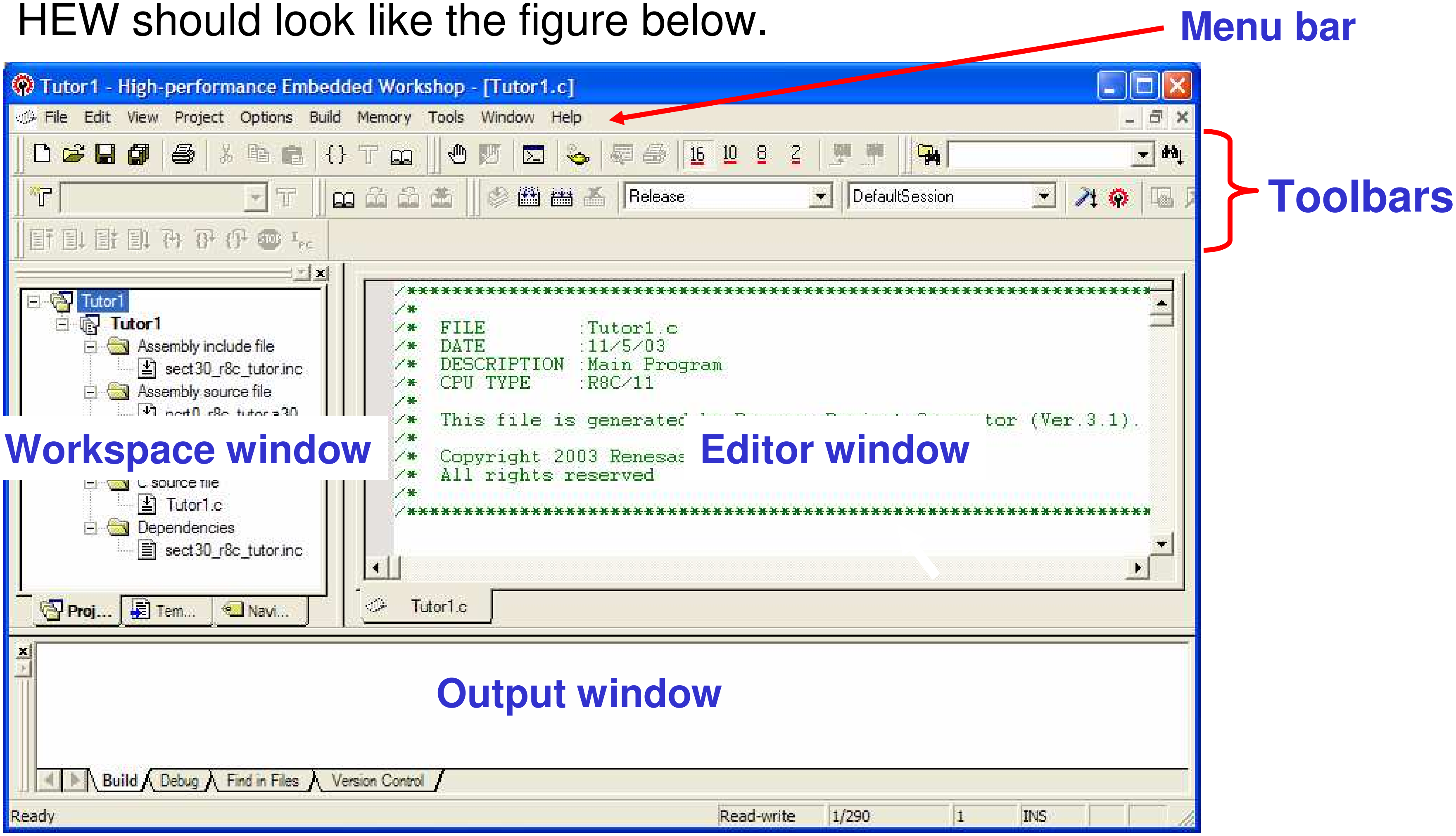
Using the Open Workspace dialog box, browse until you get to 'C:\MTOOL\SKP8CMINI\Sample_Code\Tutor1' folder. Click on Tutor1.hws HEW workspace file and then click on 'Open' button.



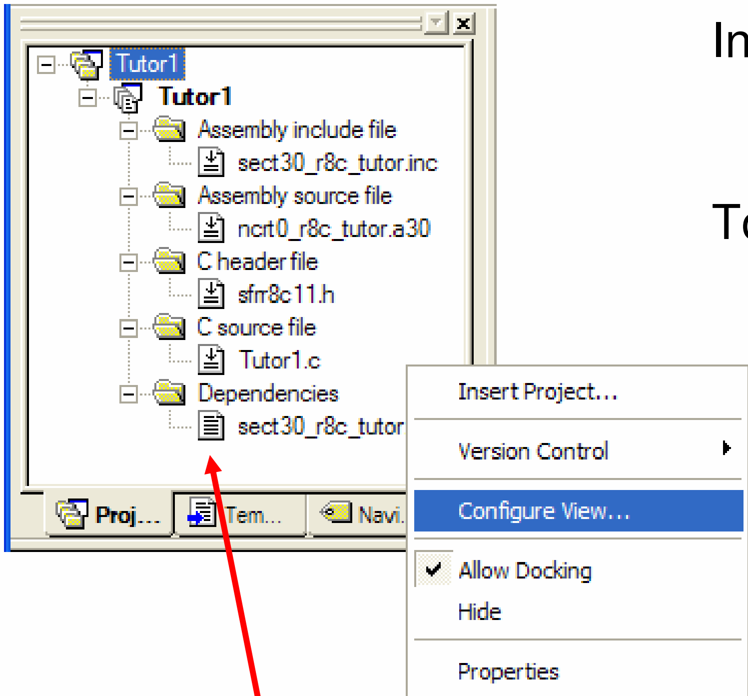
RENESAS

Open a HEW Workspace (3/3)

HEW should look like the figure below.



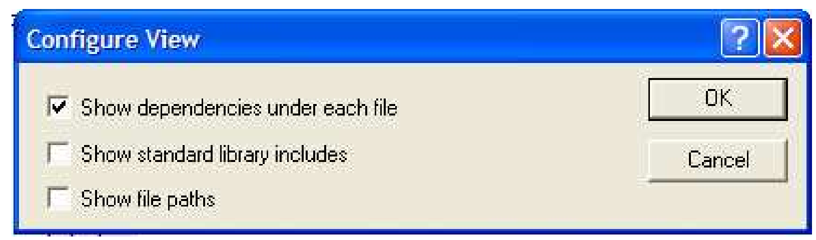
Workspace Window



To open a source file, double-click on it.

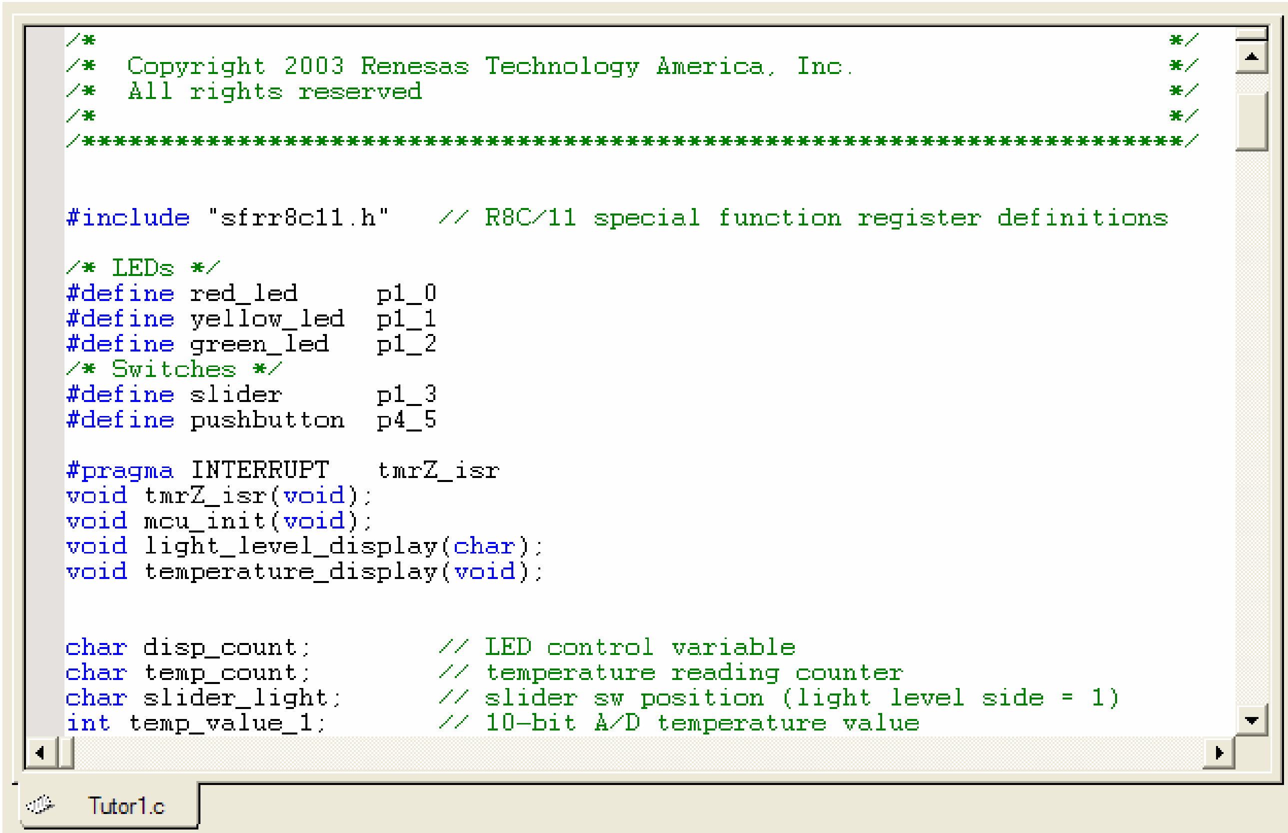
In the Projects tab, source files and header files are displayed.

To change how dependencies are displayed, e.g. show dependencies for each source file, right-click within the window, and select Configure View.



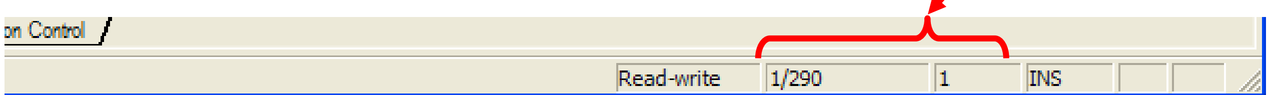
Try the following, click on 'Show dependencies under each file' and see what happens to files displayed on the window.

Editor (Source) Window



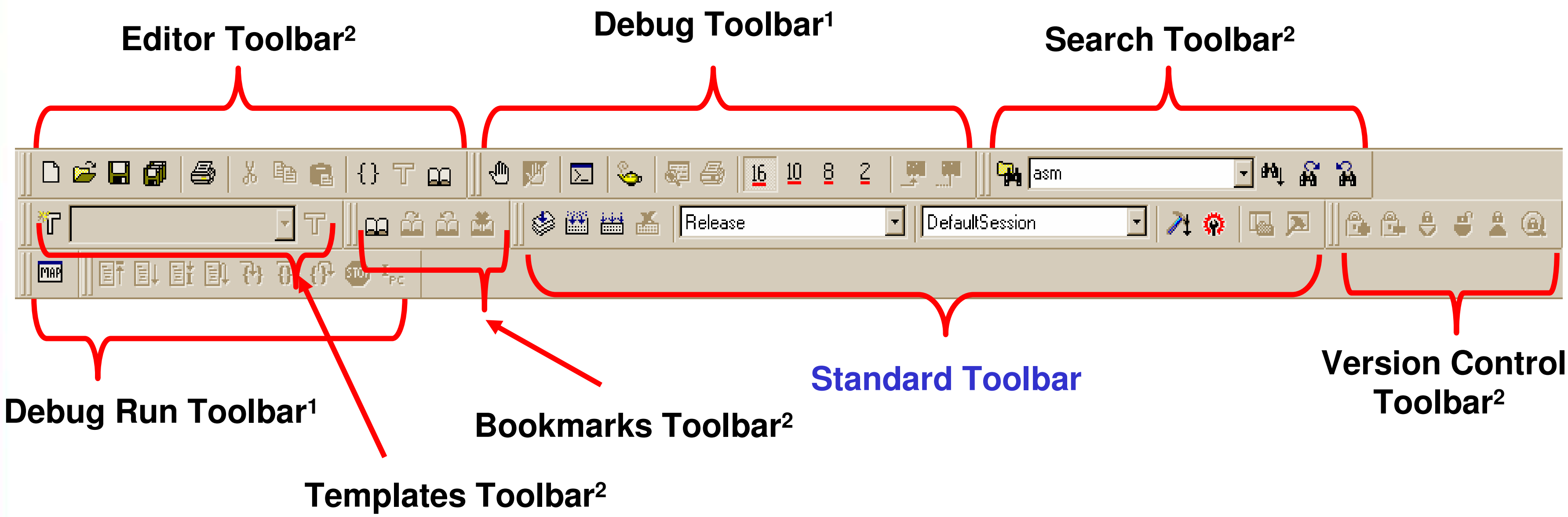
Any opened source file within the workspace are shown on the Editor window.

Line, total no. of lines, and column numbers are displayed here



HEW Toolbars

HEW is a powerful development environment with a lot of features and functionality. For this tutorial, the focus will be on features (i.e. Standard Toolbar) that will help you understand the R8C development process using HEW.

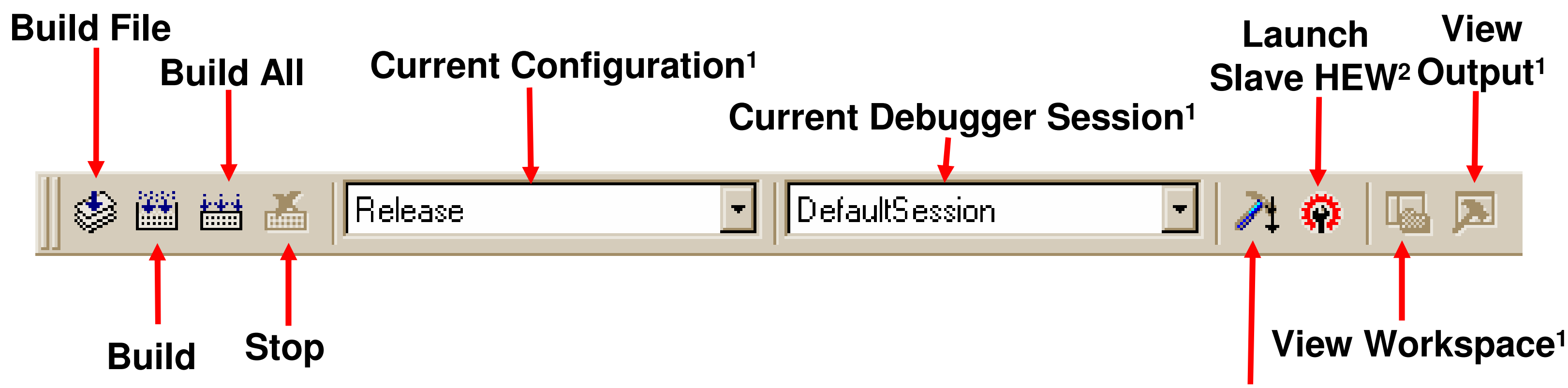


Notes:

1. ***On HEW 3.0 R1, R8C is not supported by the Debug and Debug Run toolbars.***
2. ***See HEW user's manual about these toolbars.***

RENESAS

Standard Toolbar



Build File – builds[3] a specific file
Build – builds files that were modified since last build
Build All – builds the whole project regardless of whether there were modifications or not
Stop – stops a running build process
Current Configuration – build configuration (e.g. for debug, optimized, etc)
Current Debugger Session – debug session configuration
Launch Debugger – calls defined debugger

Notes:

1. ***These features are not currently supported for R8C development.***
2. ***See HEW User's manual for details.***
3. ***A 'build' means running certain files (e.g. source files) under some tools (e.g. compiler, linker) to produce an output file (e.g X30 or MOT executable files for R8C)***

Build(re-build) Tutor1



Let's rebuild the Tutor1 project into an executable module, click on the 'Build All' icon. This will re-compile and link all the source files.

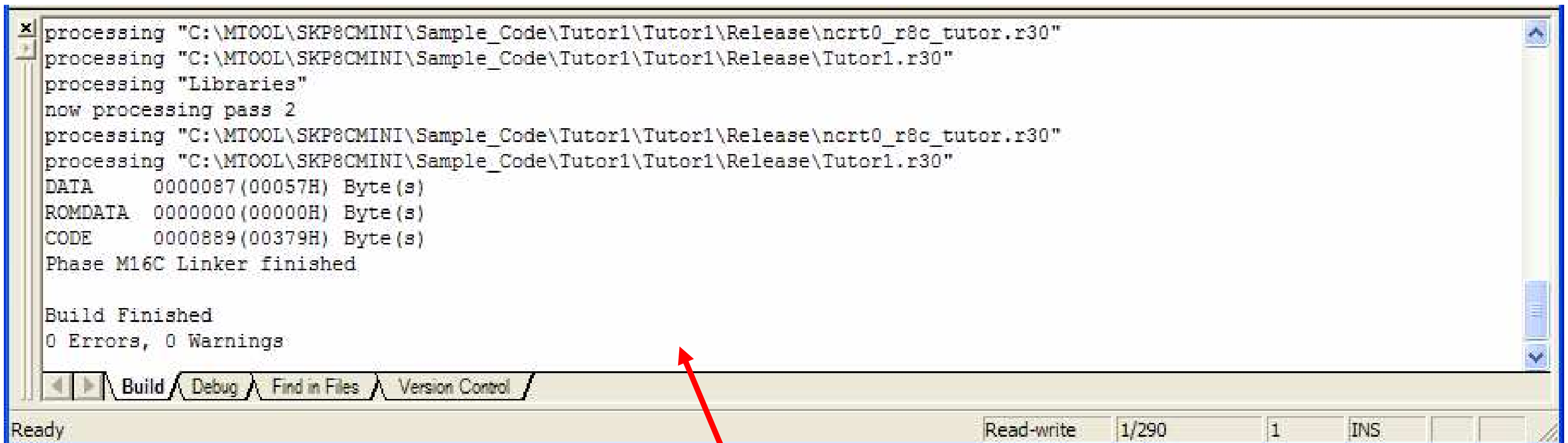
If any of the source files are modified, click on the 'Build' icon as this will only compile these modified files, which makes generating an executable module faster.

Always perform a 'Build All' when the configuration changed.

Status, errors, messages, etc during a build process is displayed on the Output window...

Output Window

The major use of the Output window is to determine if any errors or warnings occurred, and where, during the build process.



The no. of errors and warnings will show up in this window. You can then scroll up to find where the error(s) occurred. If no errors or warnings were found, ‘Build Finished’ will be displayed.

Now that an executable file has been created, the next step is to download and run the program on the Mini R8C board using the KD30 Debugger + ICD…

Do not close HEW yet. We will be returning to it later.

KD30 Debugger Overview

The KD30 Debugger can be used to verify that the program we developed works exactly as we intended and when it does not, we can also use KD30 to find out why.

Two breakpoints can be set in KD30 to stop the program at certain points (of our program) so we can verify that up to that point, the program still works correctly using registers or variables in memory.

KD30 allows “step” execution in our program, which means program execution on a per line basis (whether in source level or machine code level).

Various windows in KD30 allow us to see register values and memory locations.

KD30 Debugger Exercise

- Download and run a program on the Mini R8C board
- General use of the KD30 Debugger including stepping and setting breakpoints
- Return to HEW, modify the program, rebuild, and run the updated program on the Mini R8C board

Connect Hardware

Before starting KD30, connect the ICD to the Mini R8C board as shown. Connect the USB cable to the PC. On the ICD, the Power LED is on and the Status (Yellow) LED is blinking once a second (this means that the ICD USB driver was loaded correctly by Windows™).



To PC

USB cable

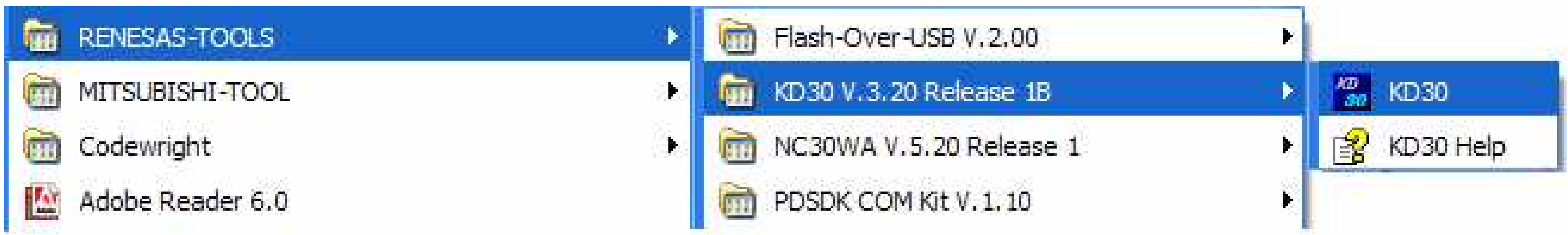
ICD (In-Circuit Debugger)

Mini R8C board

Note: The Mini R8C board connector is not keyed, so pay close attention when connecting to the ICD.

Start KD30

Launch KD30 from the Windows Start Menu,



or from HEW's Standard Toolbar[1].

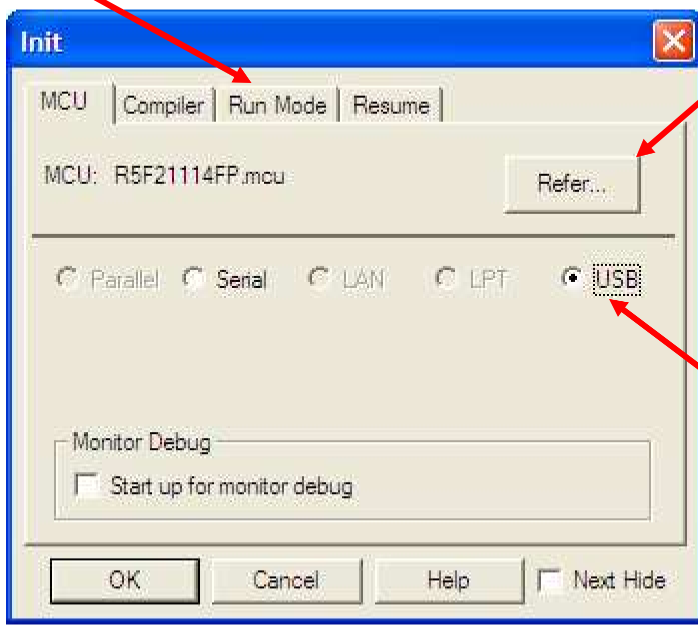


Note: 1. To call KD30 from HEW requires some configuration that is discussed in tutorial 2, Creating a New Project.

KD30 Init Window (1/2)

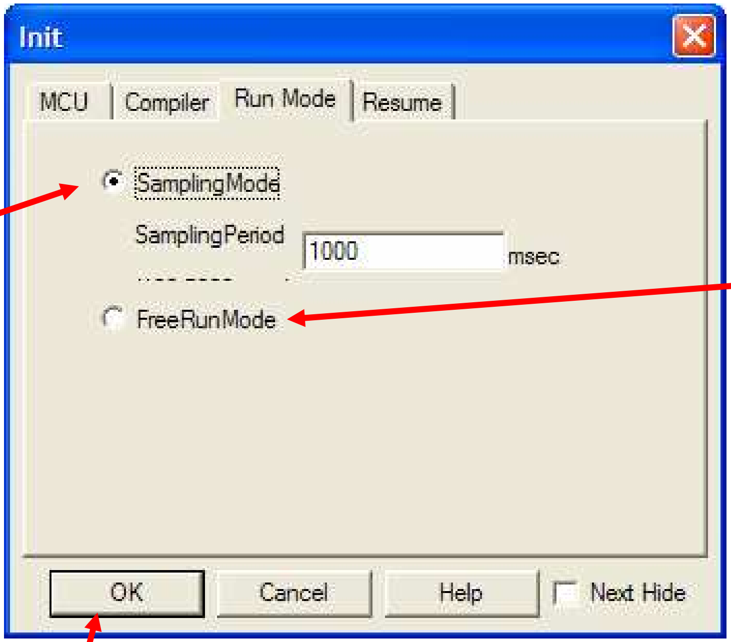
Step 3. Now click the ‘Run Mode’ tab

Step 1. Click on ‘Refer..’ and select ‘R5F21114FP.MCU’.



Step 2. Select USB

KD30 Init Window (2/2)



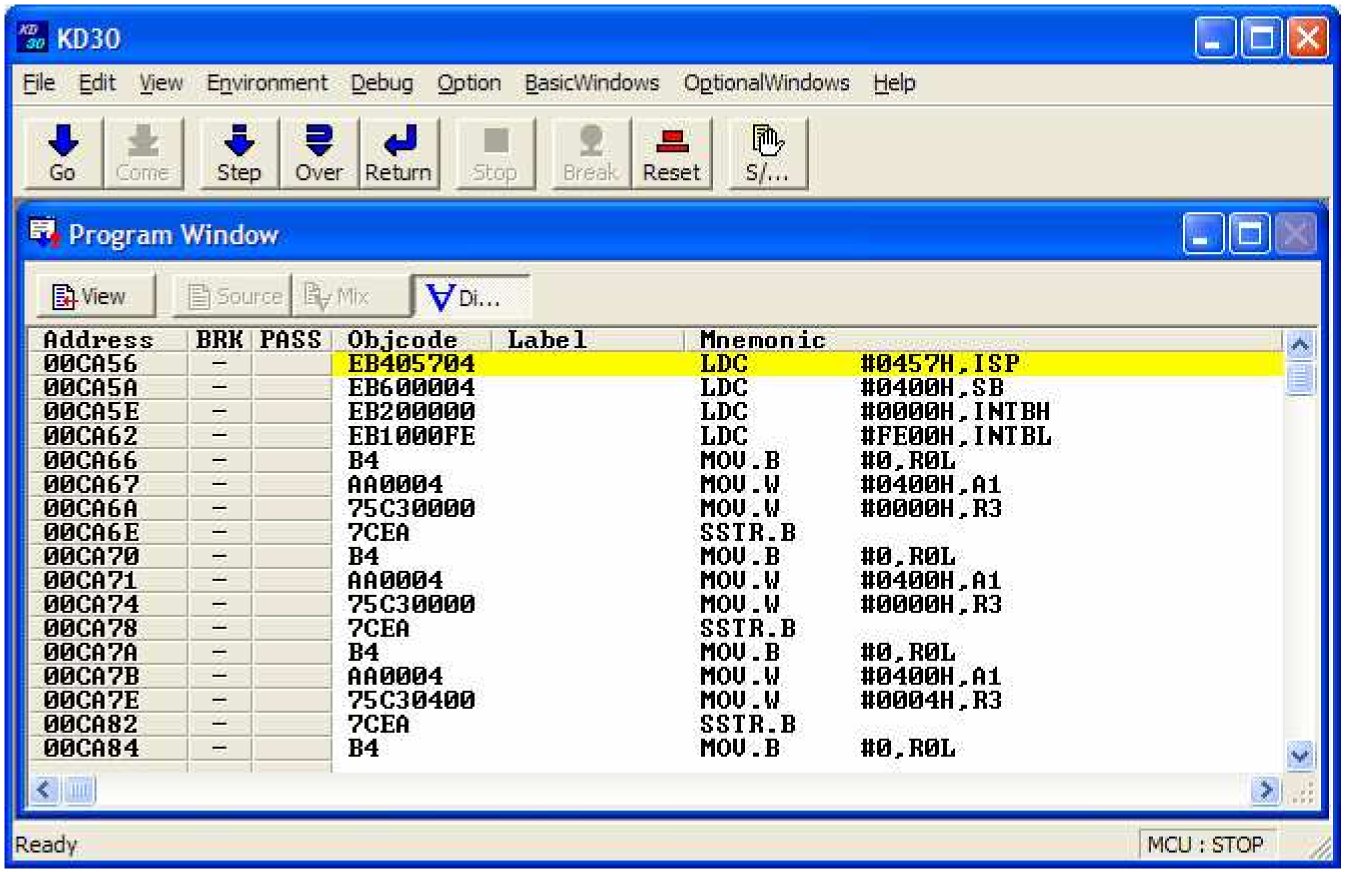
For full debugging features, be sure 'Sampling Mode[1]' is selected.

'Free Run Mode[1]' is for real time execution of your program, but debugging is limited. Do NOT select for this tutorial.

Now click 'OK' to open KD30's Program window (be sure hardware is connected). If you get an error, check all connections. See SKP user's manual on 'Troubleshooting' for details.

Note 1. See KD30 User's Manual or Help for the differences between Sampling Mode and Free Run Mode. Also, see the ICD (RTA-FoUSB-MON) User's Manual for details on how ICD works under these two modes.

KD30 Program Window



KD30 will disassemble the flash contents or display ‘UND’ if the flash is blank.

RENESAS

KD30 Toolbar

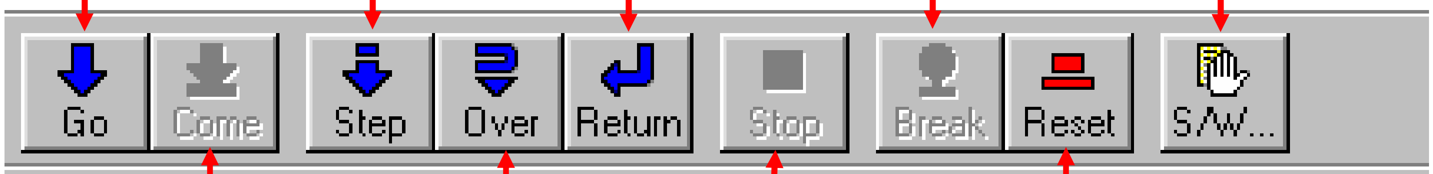
Go Button
Executes target program

Step Button
One step execution of target program

Break Button
Sets a software breakpoint at the current cursor position

Return Button
Runs the program up to the higher routine

S/W Button
Sets a software breakpoint



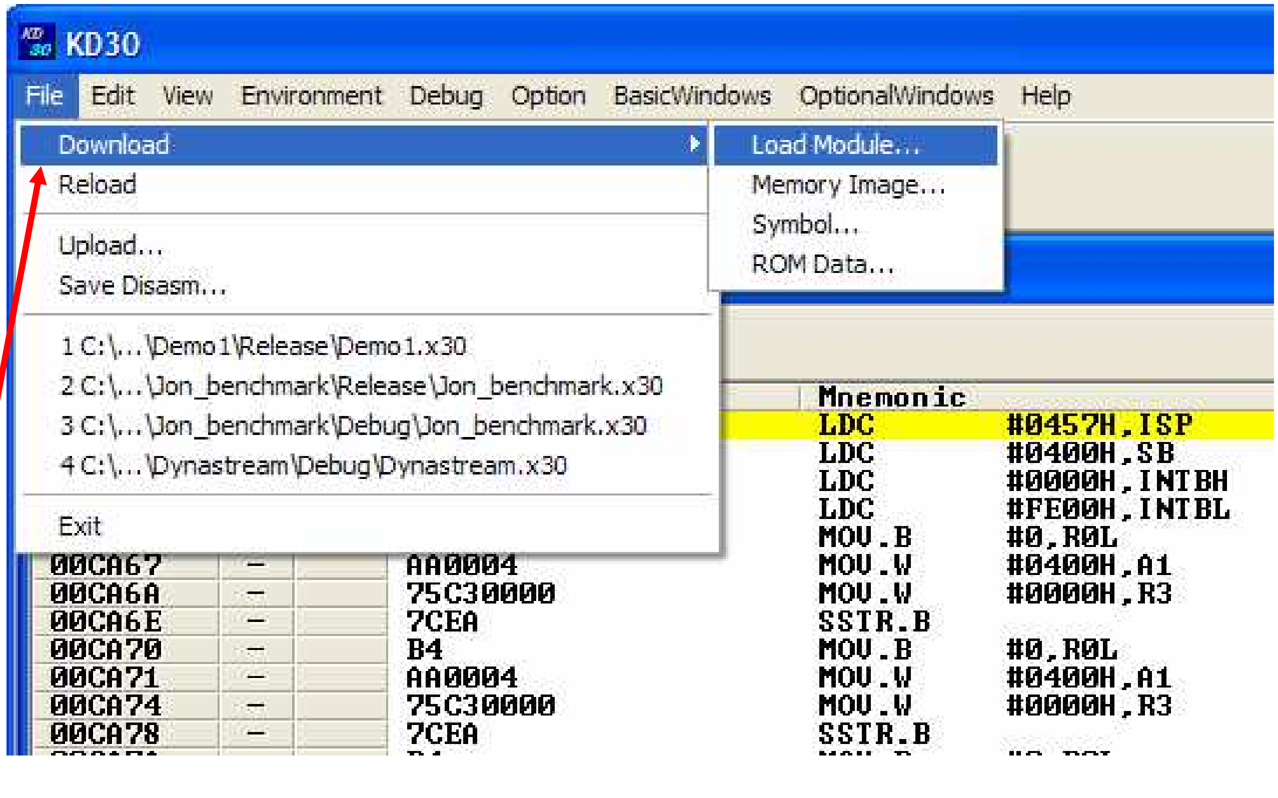
Come Button
Executes the target program from the value in the program counter to the position of the cursor in the window

Stop Button
Stops execution of the target program

Reset Button
Resets the target program

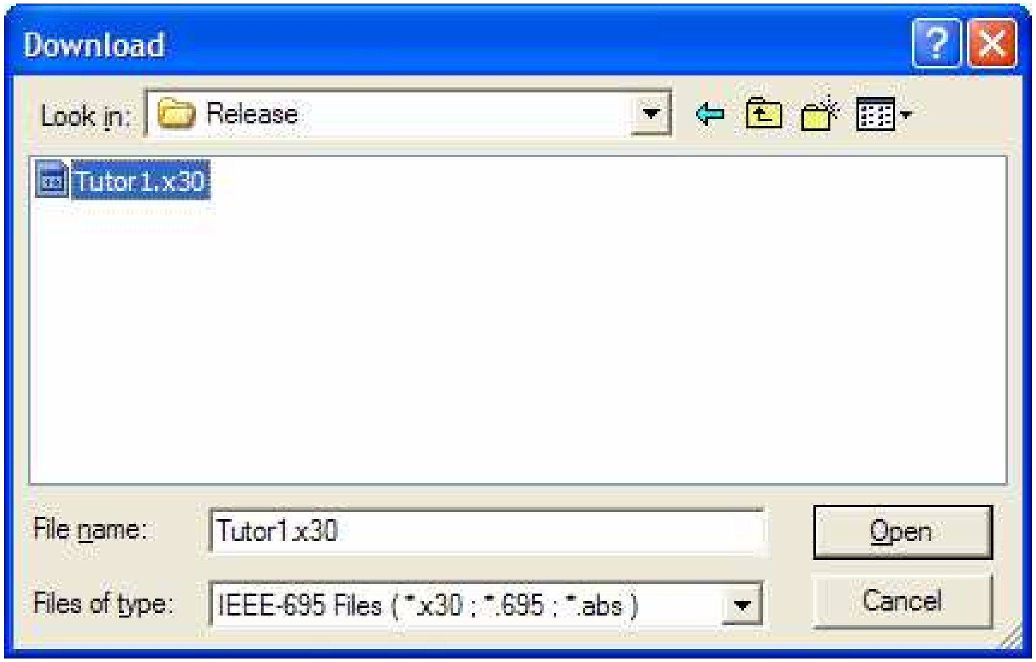
Over Button
Step over function/subroutine call

Download a Program to the Mini R8C Board (R8C/11 MCU) (1/3)



Click on 'File', then select 'Download', 'Load Module'…

Download a Program to the Mini R8C Board (R8C/11 MCU) (2/3)

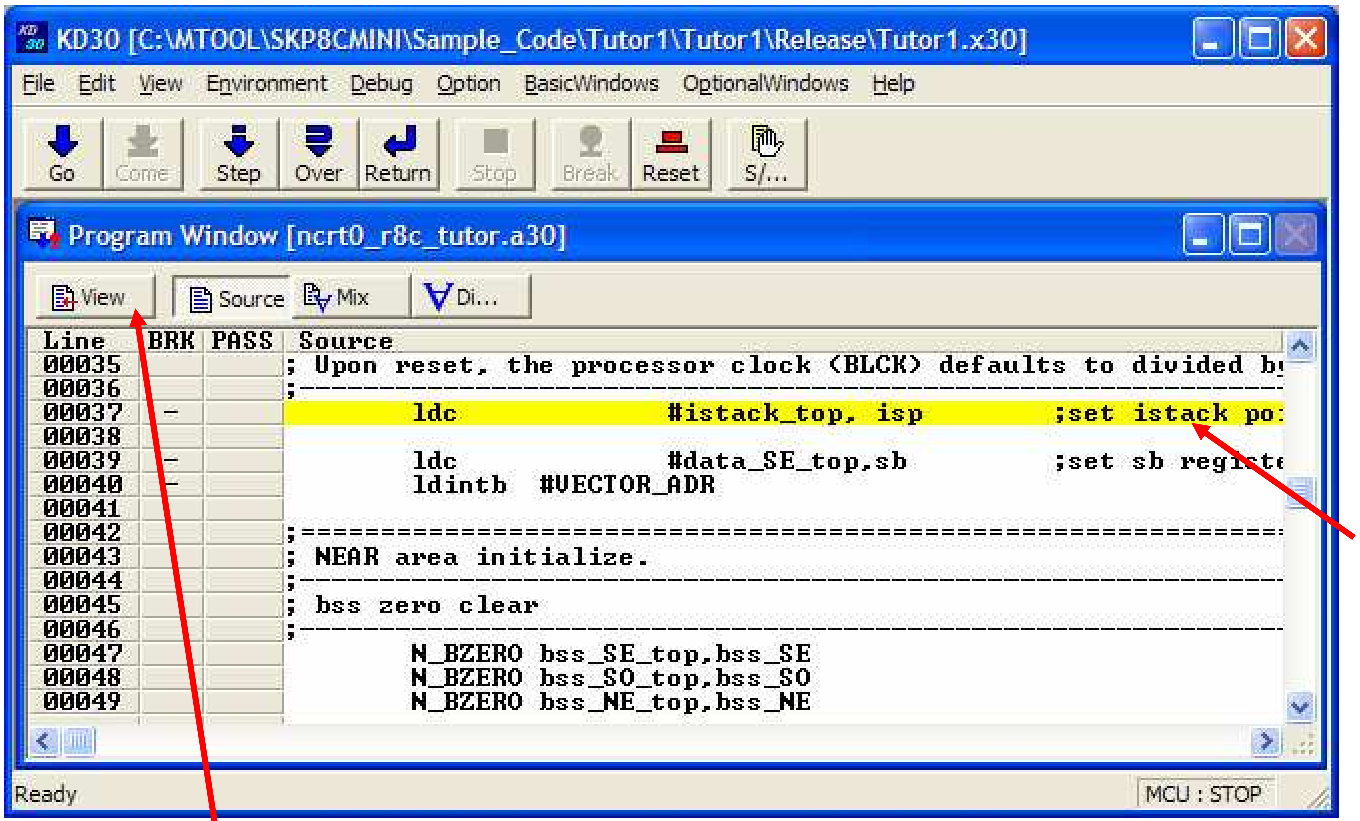


From the c:\MTOOL\SKP8CMINI\Sample_Code \Tutor1\Tutor1\release folder, select 'Tutor1.x30'.

RENESAS

Download a Program to the Mini R8C Board (R8C/11 MCU) (3/3)

After downloading the program, KD30 opens the source file where the reset vector is.



Current location of MCU program counter is highlighted.

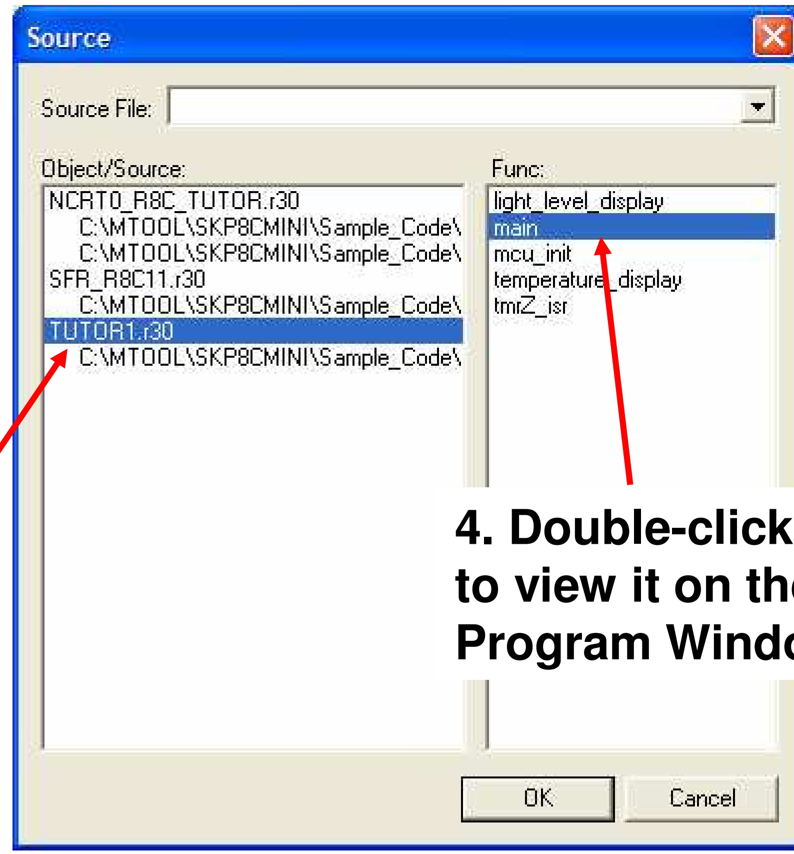
Now click on "View" to see the program source code…

Viewing Source Files in the Project

1. Click 'Source'



2. Source window is displayed.

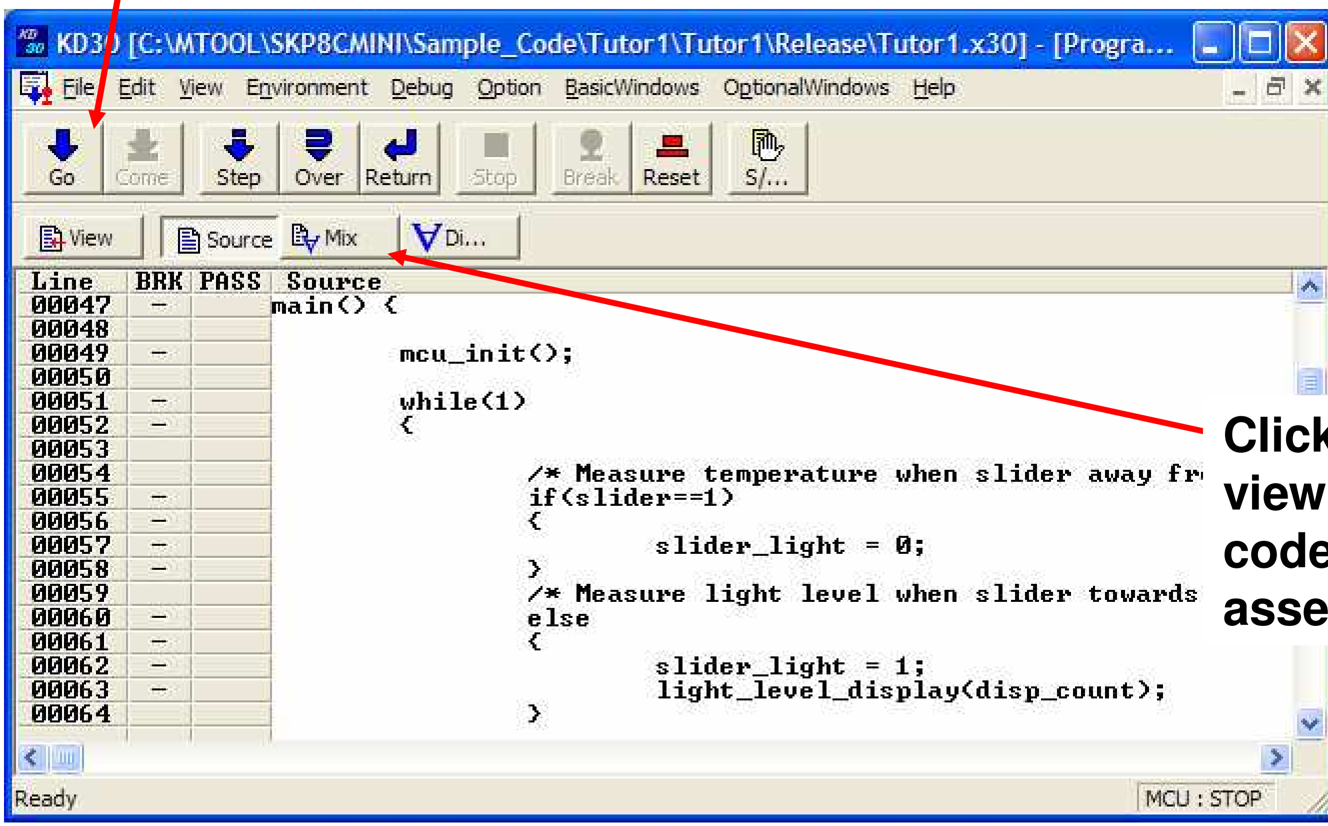


3. Click 'TUTOR1.r30'

4. Double-click 'main' to view it on the Program Window

Running Downloaded Program

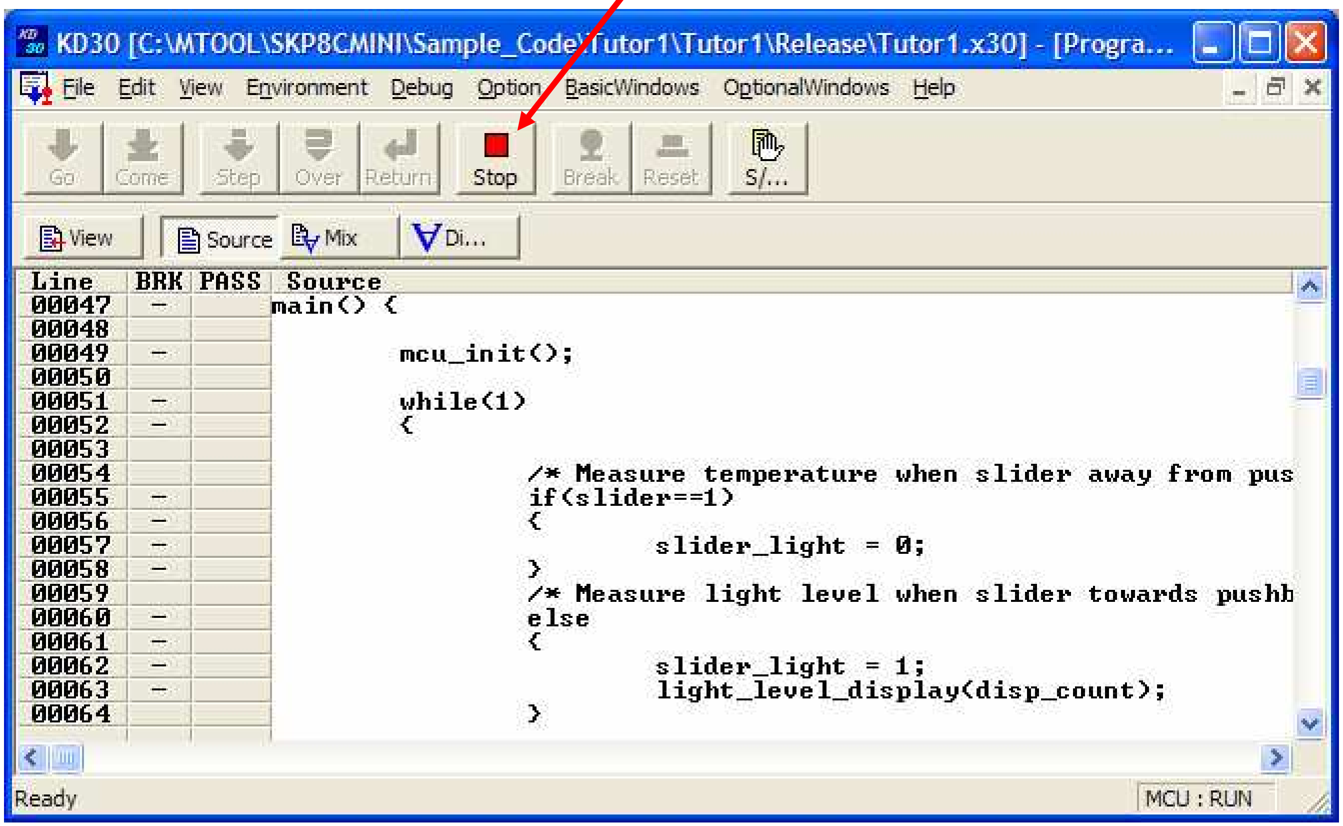
Click on the 'Go' icon to run the Tutor1 program you just downloaded. LED's D1, D2, & D3 will blink sequentially. Covering the CdS light cell will decrease the LED blink rate and uncovering it will increase it.



Click 'Mix' to view the source code and assembler code.

Stopping Program Execution

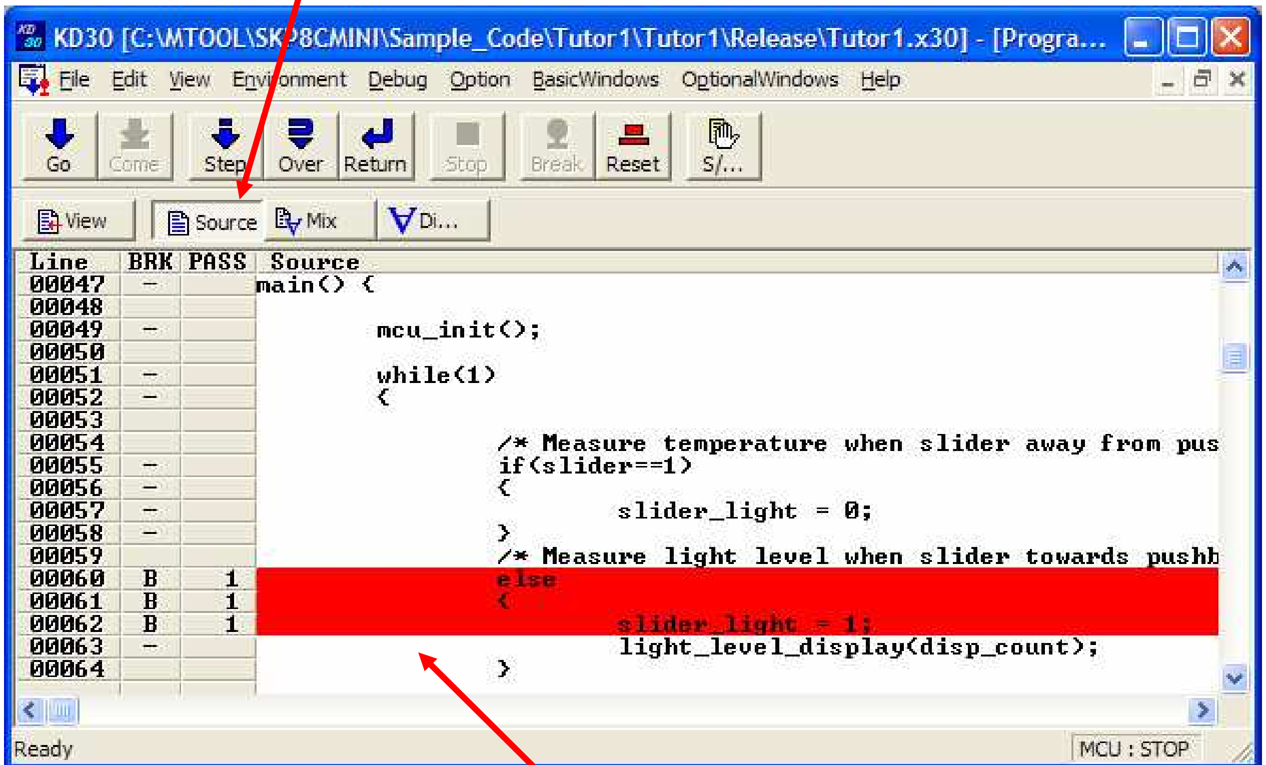
Click on the 'Stop' icon to stop the program



RENESAS

Setting Breakpoints

1. Click on the ‘Source’ to view source code only (not MIX display).

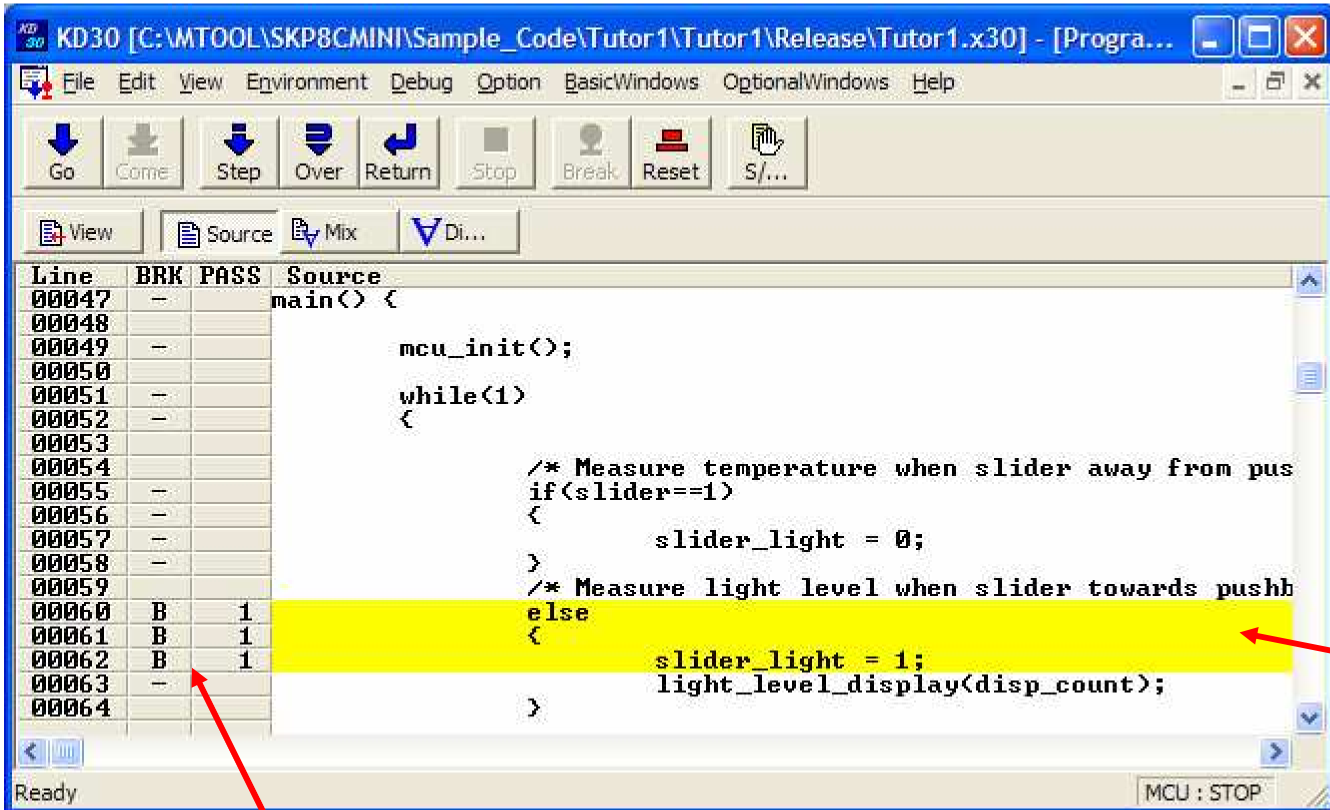


2. Locate and then set a breakpoint on ‘else’ in main() by a double-click on ‘-’ in the BRK column that denotes an executable line. A ‘B’ will appear in its place after the breakpoint is set and the line(s) is highlighted in red.

3. Click on ‘Go’ icon to run program…

RENESAS

Removing Breakpoints

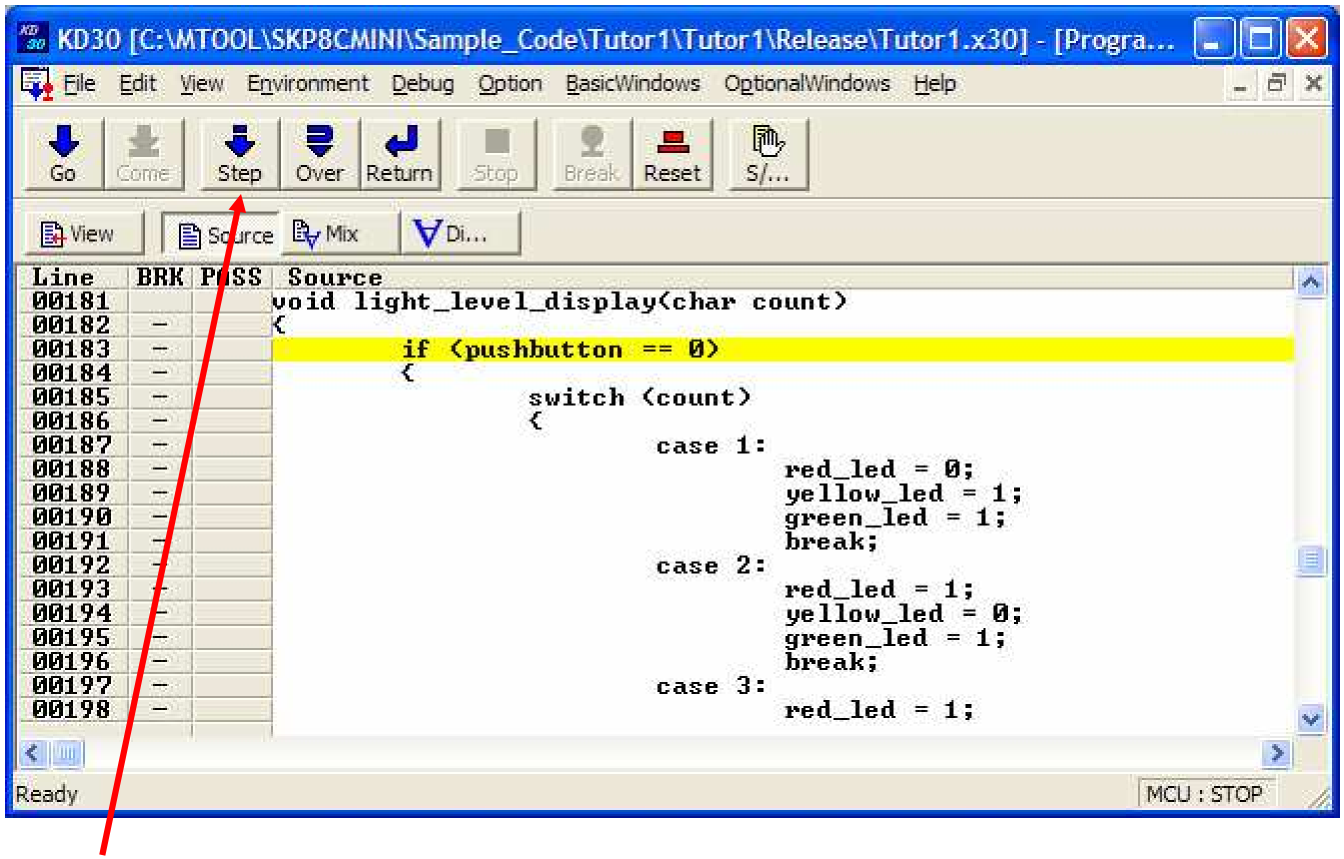


Program stops at breakpoint (highlighted in Yellow).

You can remove the breakpoint by double-clicking on it at the ‘BRK’ column.

RENESAS

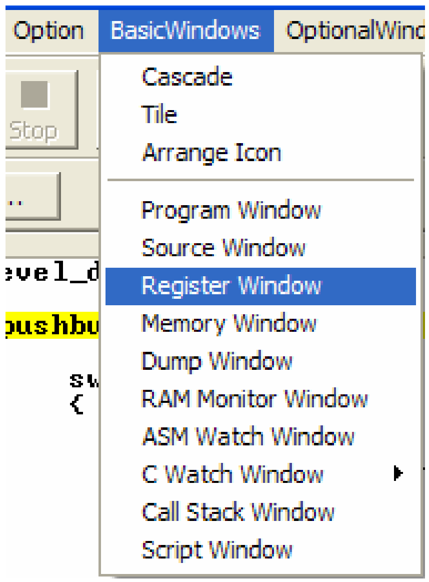
Program 'Stepping'

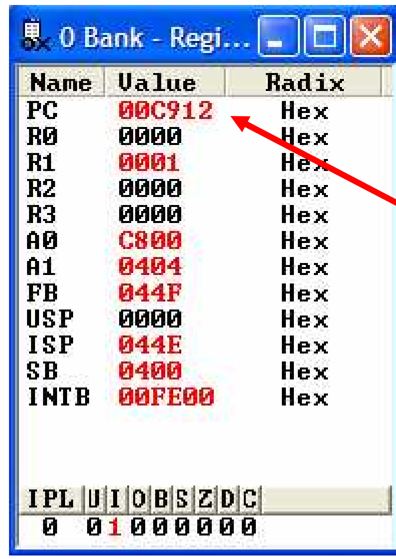


Try 'stepping' a few lines of code by clicking on 'Step' icon. Click on 'Go' afterwards to run program again.

Basic Windows: Register

Now open the ‘Register’ window





Values in red indicate changes since last “viewed”. Try ‘stepping’ and note the changes.

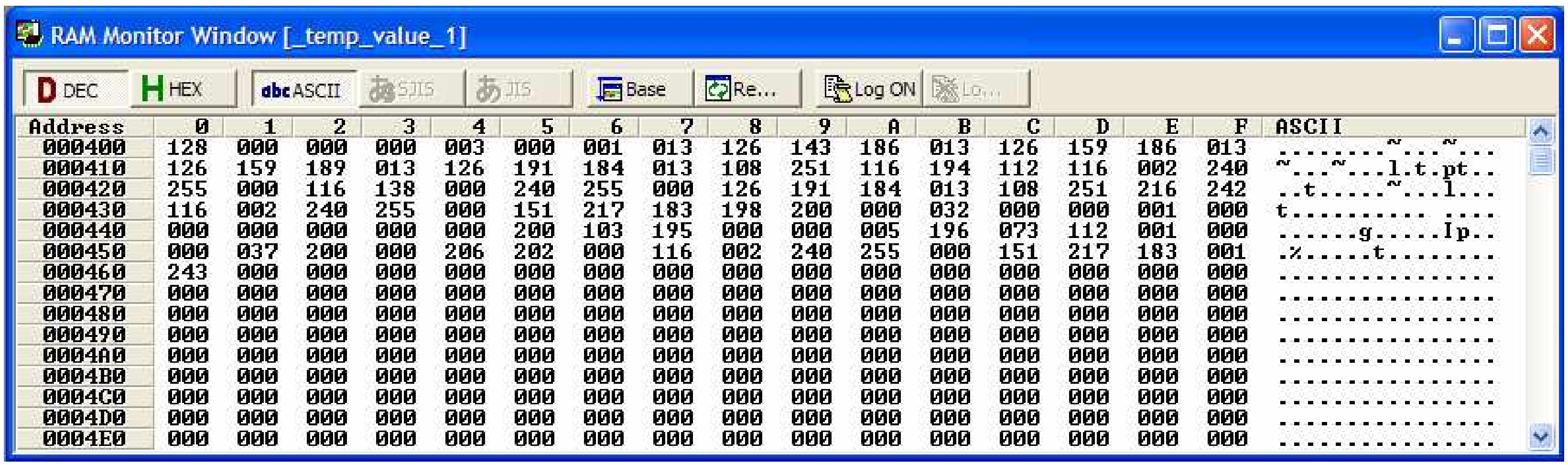
The Register window displays the values of the CPU registers after executing an instruction.

Note: Resize the Register window as needed.

Basic Windows: RAM Monitor

Open a RAM Monitor window (Basic Windows > RAM Monitor Window). The RAM Monitor displays the current value of the memory area shown on the window. It is updated at a preset value which can be modified by the user.

Double-click an address and enter 400 (hex). KD30 will tell you the page is going to change, click ‘OK’ (adjust the window size as needed).

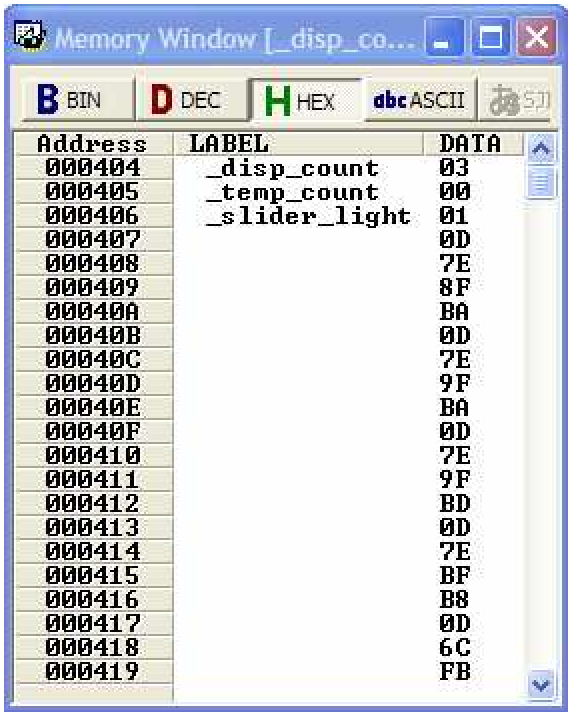


Address	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	ASCII
000400	128	000	000	000	003	000	001	013	126	143	186	013	126	159	186	013	........~...~...
000410	126	159	189	013	126	191	184	013	108	251	116	194	112	116	002	240	~...~...l.t.pt..
000420	255	000	116	138	000	240	255	000	126	191	184	013	108	251	216	242	..t.....~...l...
000430	116	002	240	255	000	151	217	183	198	200	000	032	000	000	001	000	t........... ....
000440	000	000	000	000	000	200	103	195	000	000	005	196	073	112	001	000	......g.....Ip..
000450	000	037	200	000	206	202	000	116	002	240	255	000	151	217	183	001	.%.....t........
000460	243	000	000	000	000	000	000	000	000	000	000	000	000	000	000	000	....
000470	000	000	000	000	000	000	000	000	000	000	000	000	000	000	000	000	....
000480	000	000	000	000	000	000	000	000	000	000	000	000	000	000	000	000	....
000490	000	000	000	000	000	000	000	000	000	000	000	000	000	000	000	000	....
0004A0	000	000	000	000	000	000	000	000	000	000	000	000	000	000	000	000	....
0004B0	000	000	000	000	000	000	000	000	000	000	000	000	000	000	000	000	....
0004C0	000	000	000	000	000	000	000	000	000	000	000	000	000	000	000	000	....
0004D0	000	000	000	000	000	000	000	000	000	000	000	000	000	000	000	000	....
0004E0	000	000	000	000	000	000	000	000	000	000	000	000	000	000	000	000	....

Click the ‘GO’ icon. Note you can view the RAM as it is updating. This function is not available in “Free Run” mode. Click the ‘STOP’ icon before proceeding.

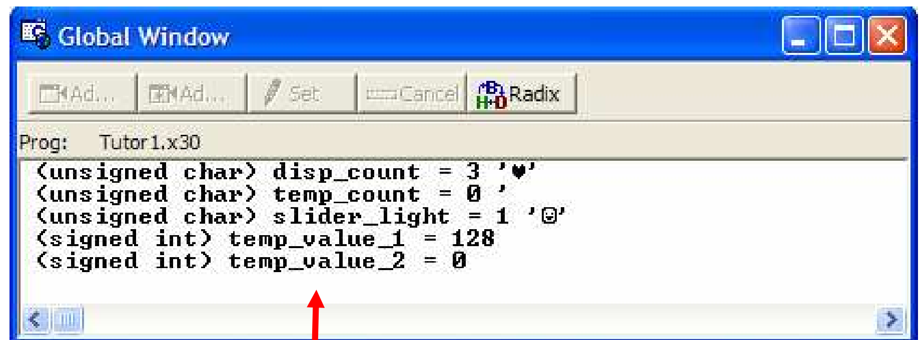
Basic Windows: Memory & C Watch

Open a Memory window (Basic Windows > Memory Window).



The 'Memory Window' displays the location and contents of variables

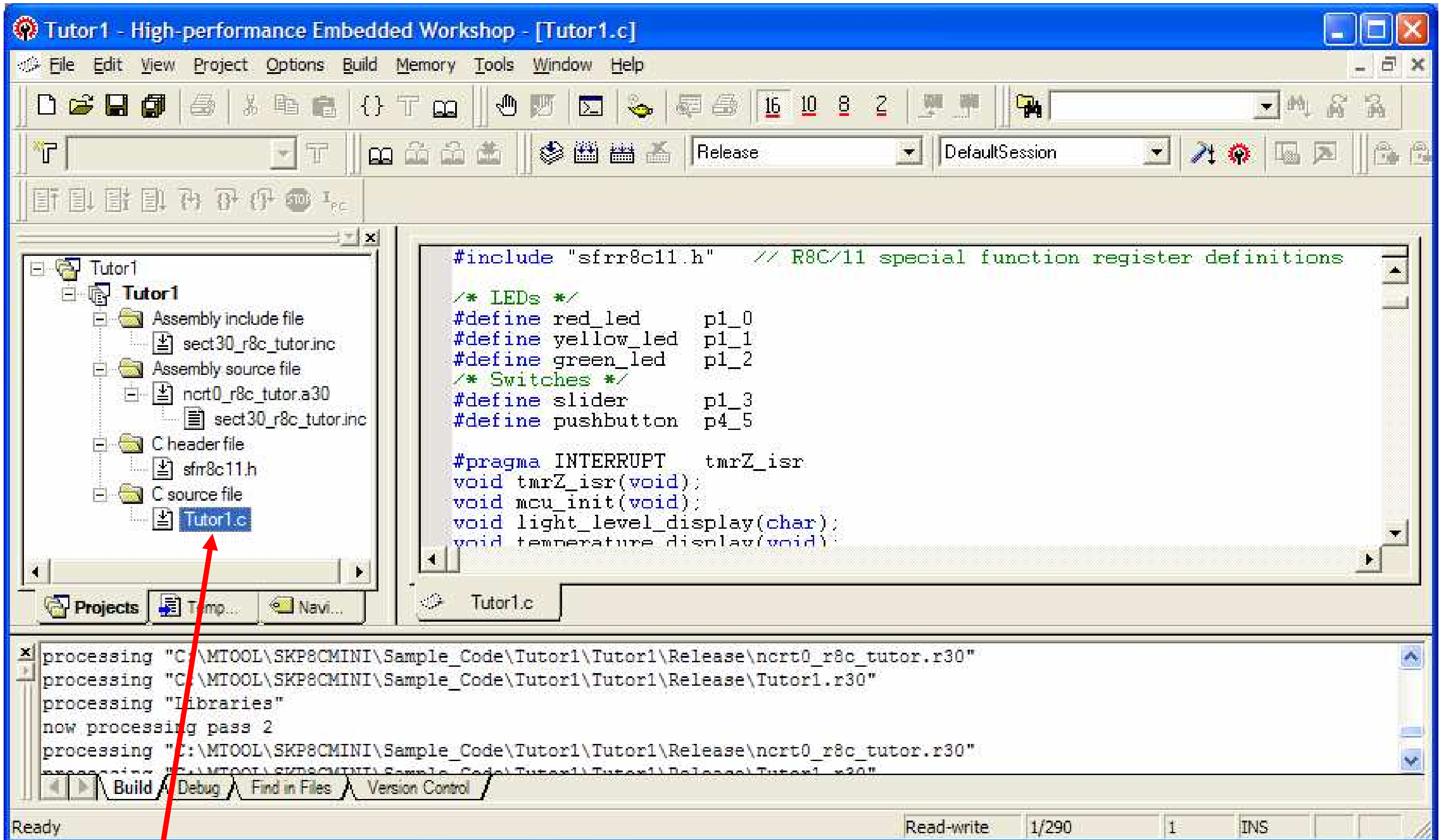
Open a C Watch window (Basic Windows > C Watch Window). The 'C Watch Window' allows you to view globals and locals. An example is shown below.



Double-click on the variable to change display format: i.e., change 'char' to 'hex' to 'decimal', etc.

RENESAS

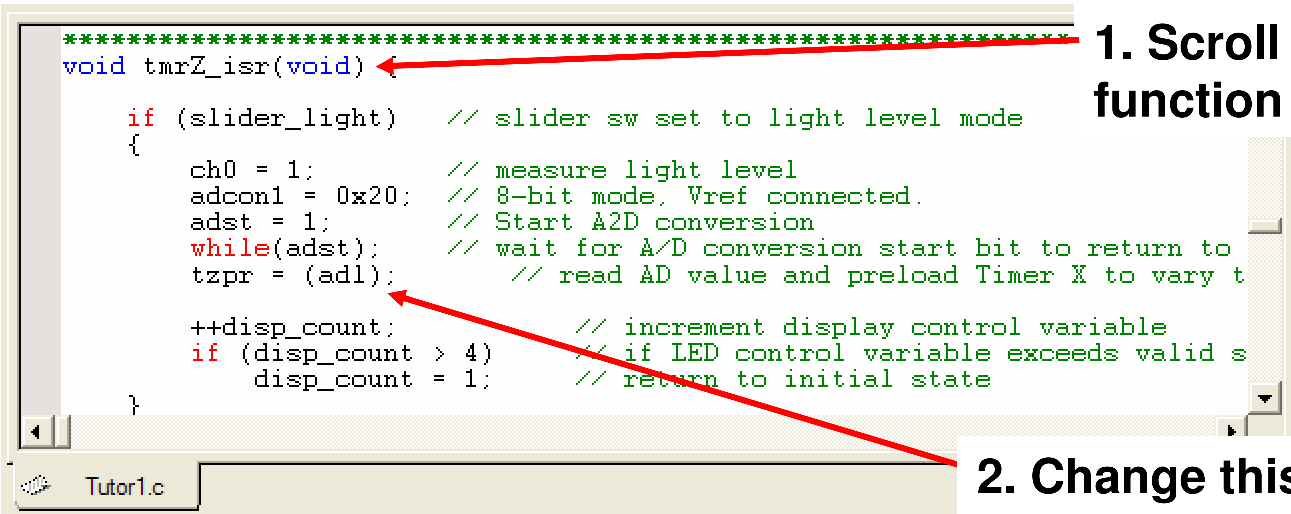
Modifying the Program (1/2)



If Tutor1.c is not shown on the Editor window, double-click on it in the Workspace window and the file will be opened/displayed on the Source window.

RENESAS

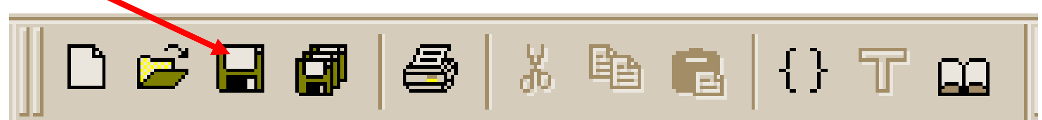
Modifying the Program (2/2)



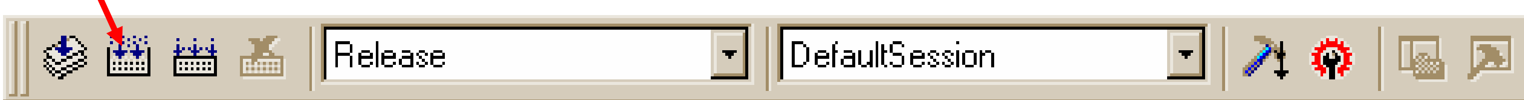
1. Scroll down and find the function ‘tmrZ_isr’ routine.

2. Change this line to ‘tzpr = (0xFF – adl);’.

3. Click this to save the revised file.

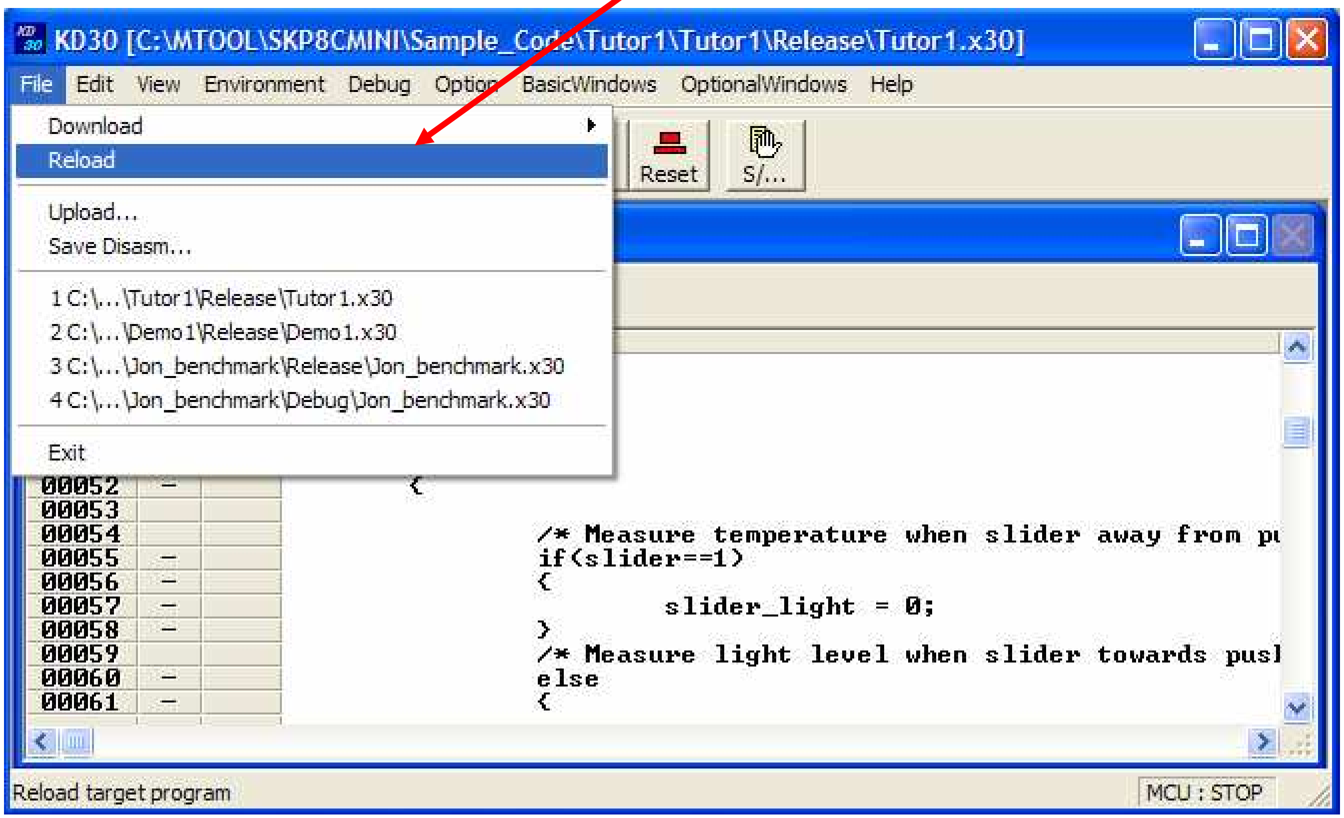


4. Build the project again.



Load (re-load) Modified Program

In KD30, with the program stopped, reload code by selecting 'Reload' from the File menu.



Covering the Light sensor on Mini R8C board increases the LED blink rate. Uncovering it decreases the blink rate.

End of Tutorial

This is the end of the tutorial. You can try downloading other sample programs from the \Sample_Code directory.

Tutorial 2 provides step by step instructions on how to use the Project Generator to simplify project creation. It also provides specific details on setting up your environment and creating a new project from scratch.

In addition, check out the references on the next page.

Have Fun!!

References and Recommended Reading

All documents that came with the SKP can be found using the “Document Description” from the Start > Programs > Renesas-Tools > SKP8CMINI menu.

- **SKP8CMINI User’s Manual:** This is a “must read” document! It details all the things you need to know on how to use the Starter Kit.
- **R8C/11 Datasheet and Mini R8C Board Schematic:** These are required to write user application programs.
- **HEW User’s Manual:** To fully understand and get the most out of HEW, this is recommended reading.
- **KD30 Version X.XX Help:** The tutorial only covered the basics of KD30. Check out the Help menu to find out all of KD30’s features.
- **NC30 Version X.XX User’s Manual:** Check this manual out for features specific to the NC30 compiler.
- **RTA-FoUSB-MON User’s Manual:** Read this manual to understand how the ICD works.

More References and Recommended Reading

- **M16C/10/20/60 Series C Language Programming Manual:** This is a great document for any level of programmer. The first chapter is an intro to C programming. The next chapter explains the memory map of C programs on microcontrollers and the role of startup programs.
- **R8CTiny Series Software Manual:** This document describes the instruction set and timing information for the R8C/Tiny series MCUs.
- **AS30 Version X.XX User's Manual:** Read this manual if you plan on writing programs in Assembly or when making changes to the startup file.
- **Application Notes and Sample Programs:** Application notes and other sample programs can be accessed from Renesas Technology America's website: http://www.renesas.com.